Álgebra lineal y sus aplicaciones en robótica con MATLAB

DÍA 1

- Del Piero Flores
- Diego Palma

Vectores

Un vector $x \in \mathbb{R}^n$ es un arreglo unidimensional de n elementos denotado por:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

con $x_i \in \mathbb{R}$. En general, es común representar a los vectores como vectores columna.

Transpuesta de un vector. La transpuesta del vector columna es un vector fila que se representa como:

$$x^T = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}$$

Usando la transpuesta, un vector también se puede denotar como $x = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^T$.

Operaciones con vectores

Norma. Se define la norma L_p de un vector $x \in \mathbb{R}^n$ como:

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{\frac{1}{p}}$$

El caso particular más conocido es la norma Euclidiana, o norma L_2. Esta calcula la magnitud de un vector y se representa como $\|x\|_2$:

$$\|x\| = \sqrt{x \cdot x} = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}$$

Producto escalar. El producto escalar o producto punto de $x, y \in \mathbb{R}^n$ denotado como $x \cdot y$ o $\langle x, y \rangle$ es un número real denotado como:

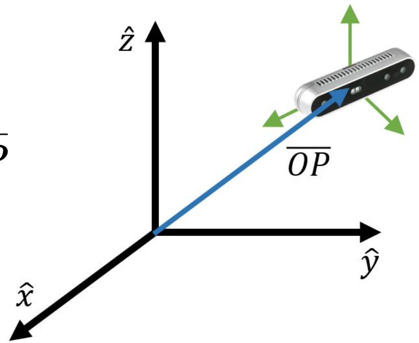
$$x \cdot y = \sum_{i=1}^{n} x_i y_i = x_1 y_1 + x_2 y_2 + \cdots + x_n y_n$$

Vector Unitario. Es aquel vector cuya norma Euclidiana es unitaria

Posición de un cuerpo rígido

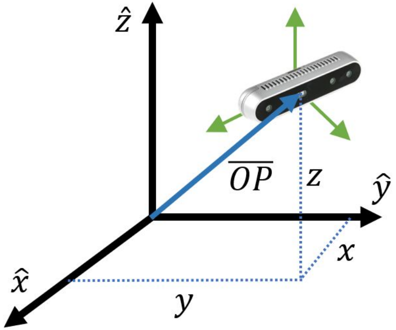
Mediante la posición de un punto P fijo al cuerpo.

Vector de posición: $\overline{OP}$



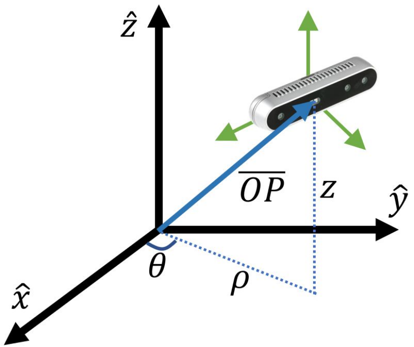
¿Cómo representamos el vector posición?

Coordenadas cartesianas



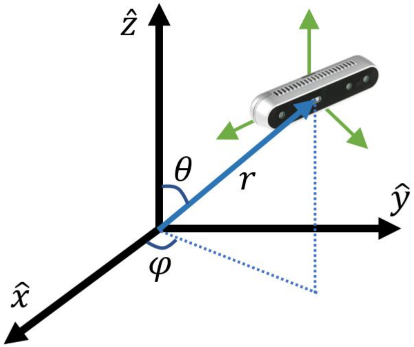
$$\overline{OP} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Coordenadas cilíndricas



$$\overline{OP} = \begin{bmatrix} \rho \\ \theta \\ z \end{bmatrix}$$

Coordenadas esféricas



$$\overline{OP} = \begin{bmatrix} r \\ \theta \\ \varphi \end{bmatrix}$$

Matrices

Una matriz $A \in R^{m \times n}$ es un arreglo bidimensional de $(m \times n)$ elementos y cuya representación es de la siguiente forma

$$A = [x_{ij}] = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}$$

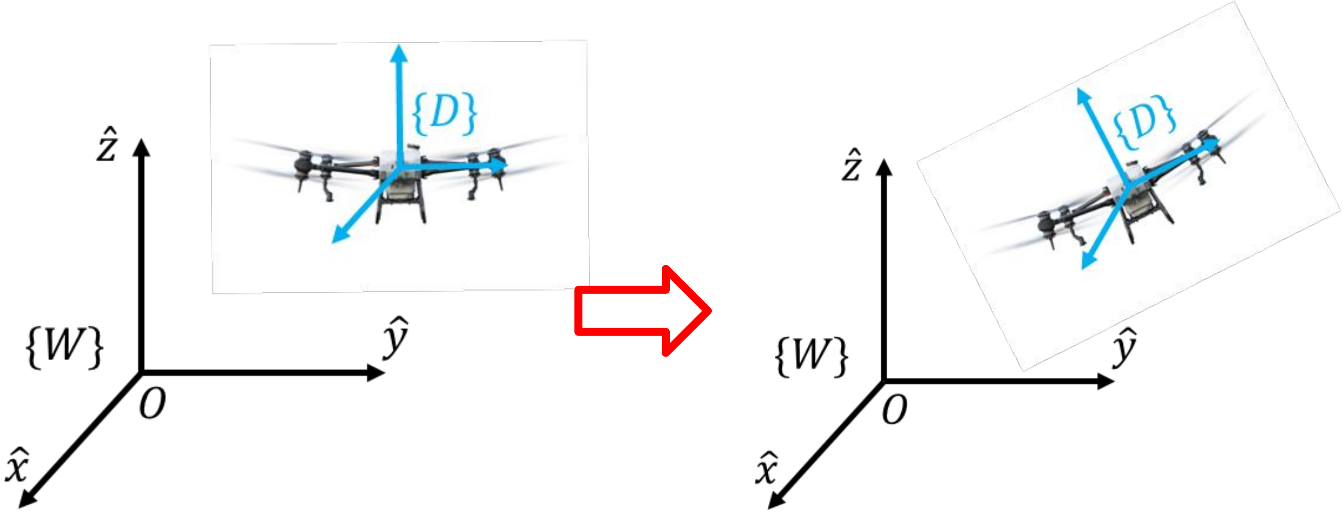
Transpuesta de una matriz

$$A^T = \begin{bmatrix} x_{11} & x_{21} & \cdots & x_{m1} \\ x_{12} & x_{22} & \cdots & x_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1n} & x_{1n} & \cdots & x_{mn} \end{bmatrix}$$

Si $m = n$, se dice que es una matriz cuadrada; de otro modo, la matriz es rectangular

Orientación de un cuerpo rígido

¿Cómo representamos la orientación de un cuerpo rígido?

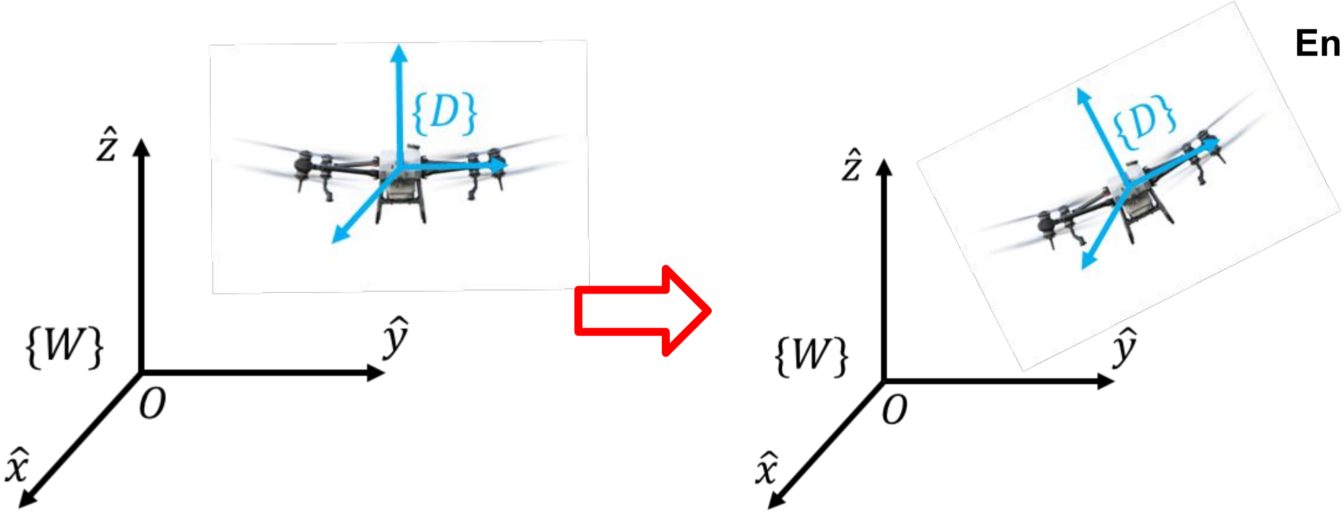


Orientación de un cuerpo rígido

¿Cómo representamos la orientación de un cuerpo rígido?

¿Cuántas coordenadas se necesitan?

En 2D:

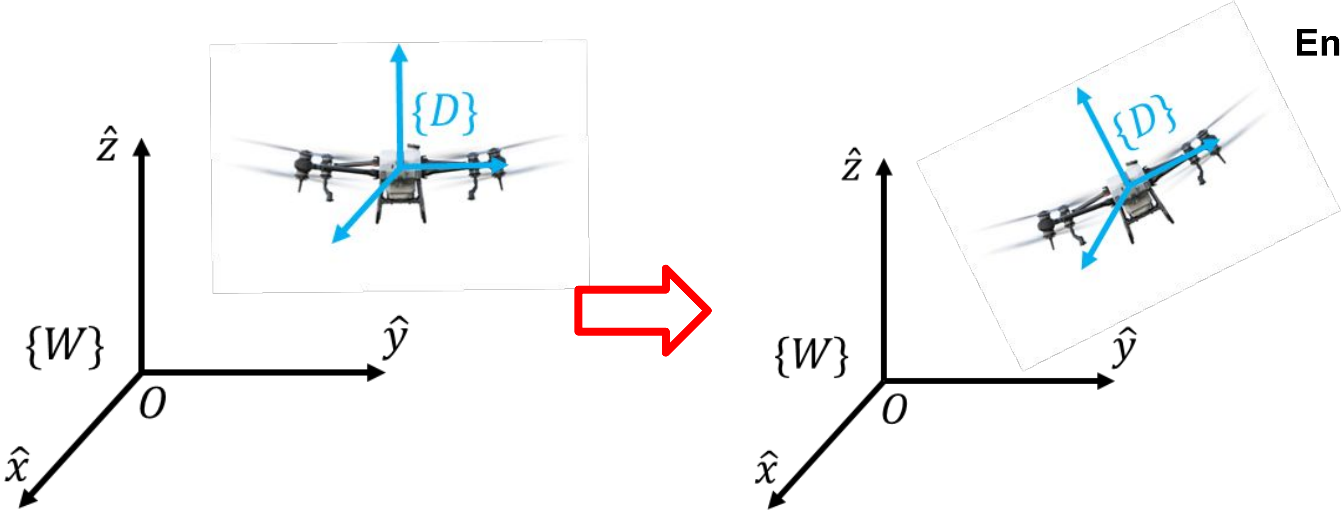


Orientación de un cuerpo rígido

¿Cómo representamos la orientación de un cuerpo rígido?

¿Cuántas coordenadas se necesitan?

En 2D: 1 coordenada (ángulo)



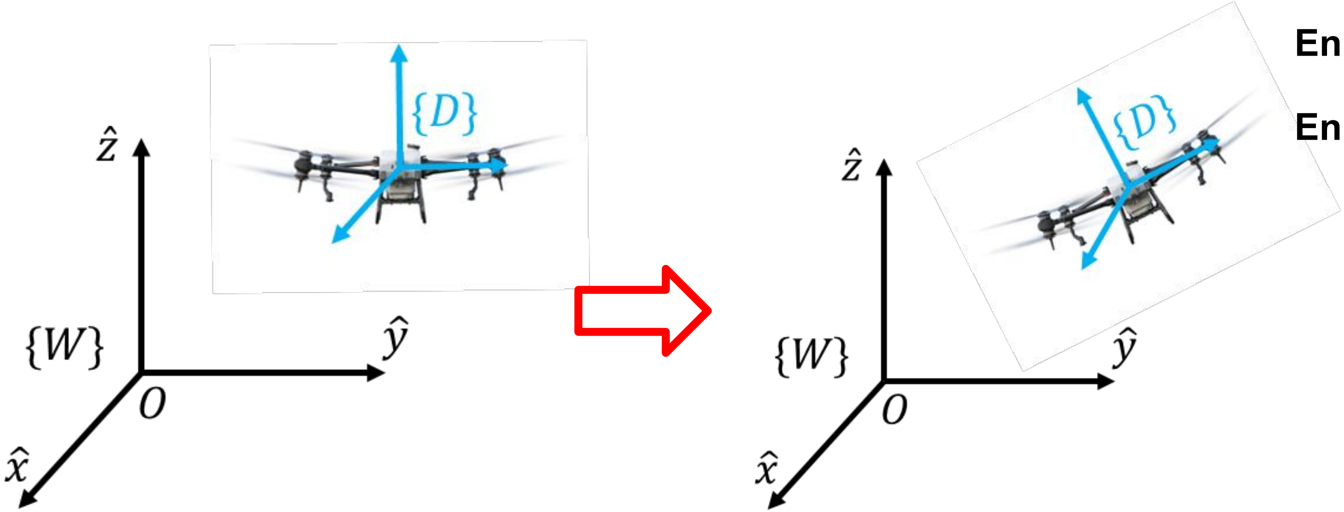
Orientación de un cuerpo rígido

¿Cómo representamos la orientación de un cuerpo rígido?

¿Cuántas coordenadas se necesitan?

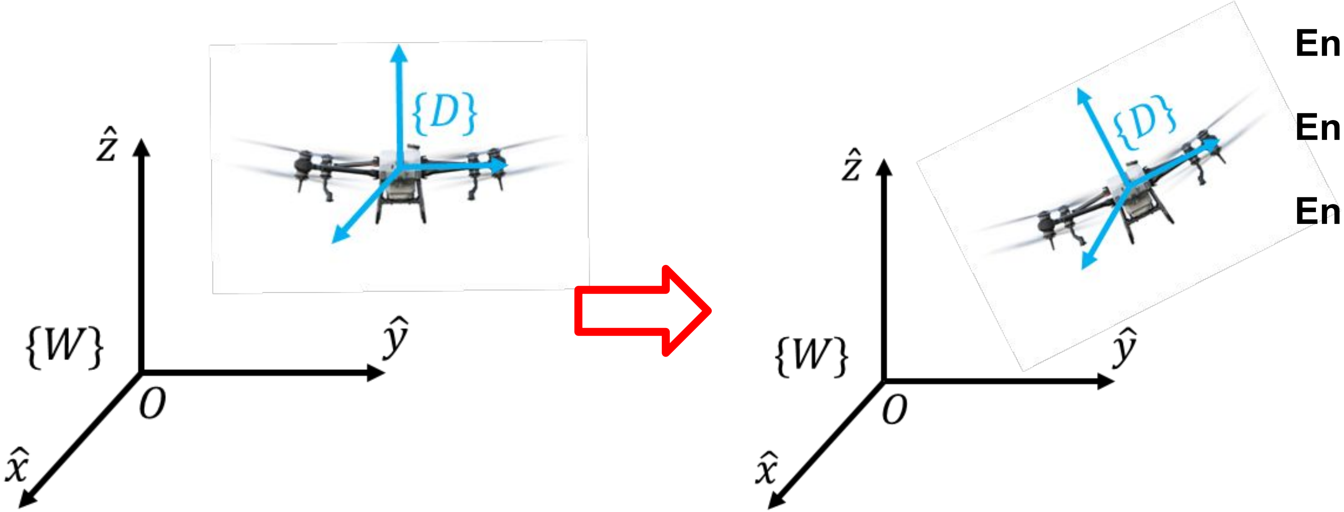
En 2D: 1 coordenada (ángulo)

En 3D: 3 coordenadas



Orientación de un cuerpo rígido

¿Cómo representamos la orientación de un cuerpo rígido?



¿Cuántas coordenadas se necesitan?

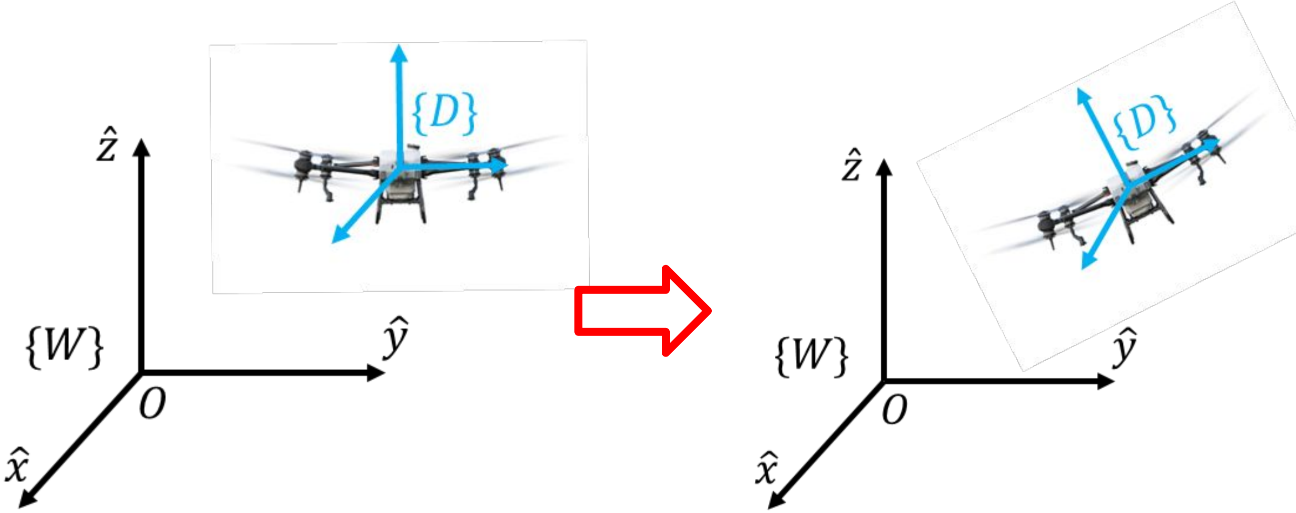
En 2D: 1 coordenada (ángulo)

En 3D: 3 coordenadas

En n-D: n(n-1)/2 coordenadas

Matriz de rotación

¿Cómo representamos la orientación de un cuerpo rígido?



¿Cuántas coordenadas se necesitan?

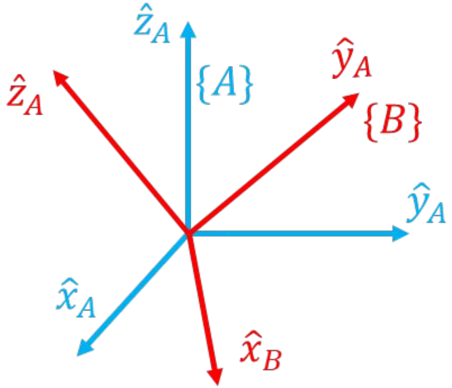
En 2D: 1 coordenada (ángulo)

En 3D: 3 coordenadas

En n-D: n(n-1)/2 coordenadas

Matriz de rotación

Permite representar las coordenadas de un sistema con respecto a otro



Proyectando los ejes de A, en cada eje de B

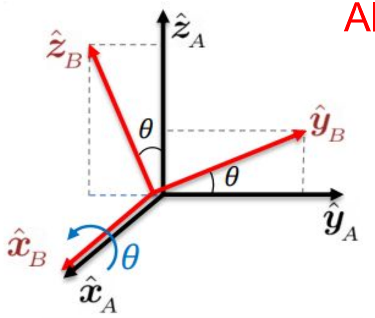
Matriz de rotación del sistema {B} con respecto al sistema {A}

$$
{}^{A}\mathbf{R}_B = [\ {}^{A}\hat{\mathbf{x}}_B \quad {}^{A}\hat{\mathbf{y}}_B \quad {}^{A}\hat{\mathbf{z}}_B] = \begin{bmatrix} \hat{\mathbf{x}}_B.\hat{\mathbf{x}}_A & \hat{\mathbf{y}}_B.\hat{\mathbf{x}}_A & \hat{\mathbf{z}}_B.\hat{\mathbf{x}}_A \\ \hat{\mathbf{x}}_B.\hat{\mathbf{y}}_A & \hat{\mathbf{y}}_B.\hat{\mathbf{y}}_A & \hat{\mathbf{z}}_B.\hat{\mathbf{y}}_A \\ \hat{\mathbf{x}}_B.\hat{\mathbf{z}}_A & \hat{\mathbf{y}}_B.\hat{\mathbf{z}}_A & \hat{\mathbf{z}}_B.\hat{\mathbf{z}}_A \end{bmatrix}_{3\times3}
$$

Matriz de rotación

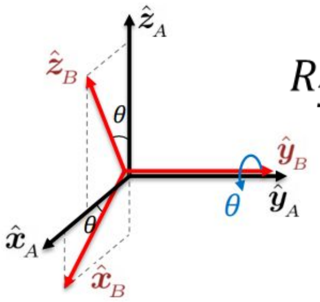
Rotaciones elementales

Alrededor del eje x



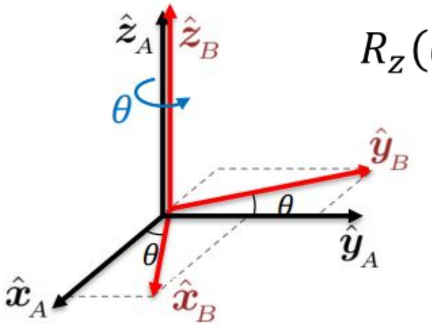
$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$

Alrededor del eje y



$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

Alrededor del eje z



$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Multiplicación de matrices y vectores

El producto de dos matrices $A \in \mathbb{R}^{m \times n}$ y $B \in \mathbb{R}^{n \times p}$ es la matriz $C = AB \in \mathbb{R}^{m \times p}$ cuyos términos c_{ij} se definen como

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

Para que el producto exista, el número de columnas de **A** debe ser igual al número de filas de **B**

Producto matriz-vector

$$A = [\boldsymbol{a}_1 \ \ \boldsymbol{a}_2 \ \ \ldots \ \ \boldsymbol{a}_n] \qquad \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$A\boldsymbol{x} = [\boldsymbol{a}_1 \ \ \boldsymbol{a}_2 \ \ \ldots \ \ \boldsymbol{a}_n] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \boldsymbol{a}_1 x_1 + \boldsymbol{a}_2 x_2 + \cdots + \boldsymbol{a}_n x_n$$

Producto matriz-matriz

$$A = \begin{bmatrix} \boldsymbol{a}_1^T \\ \boldsymbol{a}_2^T \\ \vdots \\ \boldsymbol{a}_m^T \end{bmatrix} \qquad B = [\boldsymbol{b}_1 \ \ \boldsymbol{b}_2 \ \ \cdots \ \ \boldsymbol{b}_p]$$

$$AB = \begin{bmatrix} \boldsymbol{a}_1^T \\ \boldsymbol{a}_2^T \\ \vdots \\ \boldsymbol{a}_m^T \end{bmatrix} [\boldsymbol{b}_1 \ \ \boldsymbol{b}_2 \ \ \cdots \ \ \boldsymbol{b}_p] = \begin{bmatrix} \boldsymbol{a}_1^T \boldsymbol{b}_1 & \boldsymbol{a}_1^T \boldsymbol{b}_2 & \cdots & \boldsymbol{a}_1^T \boldsymbol{b}_p \\ \boldsymbol{a}_2^T \boldsymbol{b}_1 & \boldsymbol{a}_2^T \boldsymbol{b}_2 & \cdots & \boldsymbol{a}_2^T \boldsymbol{b}_p \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{a}_m^T \boldsymbol{b}_1 & \boldsymbol{a}_m^T \boldsymbol{b}_2 & \cdots & \boldsymbol{a}_m^T \boldsymbol{b}_p \end{bmatrix}$$

Propiedades de una Matriz de Rotación

- La multiplicación de una matriz de rotación y su transpuesta da como resultado la matriz identidad.

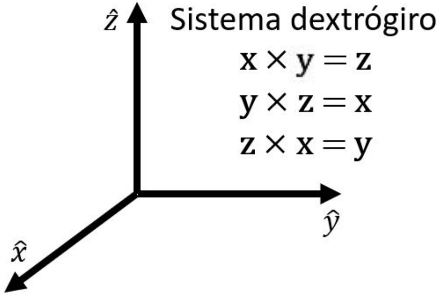
$$RR^T = \begin{bmatrix} - & \hat{x}^T & - \\ - & \hat{y}^T & - \\ - & \hat{z}^T & - \end{bmatrix} \begin{bmatrix} | & | & | \\ \hat{x}^T & \hat{y}^T & \hat{z}^T \\ | & | & | \end{bmatrix} = \underbrace{\begin{bmatrix} \hat{x}^T\hat{x} & \hat{x}^T\hat{y} & \hat{x}^T\hat{z} \\ \hat{y}^T\hat{x} & \hat{y}^T\hat{y} & \hat{y}^T\hat{z} \\ \hat{z}^T\hat{x} & \hat{z}^T\hat{y} & \hat{z}^T\hat{z} \end{bmatrix}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Cada eje $(\hat{x}, \hat{y}, \hat{z})$ es unitario y perpendicular a los otros 2 ejes. El producto escalar de dos vectores perpendiculares es 0.

Por ende, la transpuesta de una matriz de rotación es la inversa de dicha matriz. $RR^T = I \Rightarrow R^T = R^{-1}$

- La determinante de una matriz de rotación es ± 1

$$\begin{aligned} R^T R &= I \\ \det\left(R^T R\right) &= \det(I) \\ \det\left(R^T\right)\det(R) &= 1 \\ (\det(R))^2 &= 1 \\ \det(R) &= \pm 1 \end{aligned}$$



$$\det(R) = +1$$

Relaciones y composición de rotaciones

- Considerando un punto en 3 sistemas: {A}, {B} y {C}
  - Punto ***P*** en el sistema {A}: ${}^{A}p$
  - Punto ***P*** en el sistema {B}: ${}^{B}p$
  - Punto ***P*** en el sistema {C}: ${}^{C}p$
- Relaciones:
  - Punto del sistema {B} en el sistema {A}: ${}^{A}p = {}^{A}R_B\, {}^{B}p$
  - Punto del sistema {C} en el sistema {B}: ${}^{B}p = {}^{B}R_C\, {}^{C}p$
  - Punto del sistema {C} en el sistema {A}: ${}^{A}p = {}^{A}R_C\, {}^{C}p$
- Composición de rotaciones:

$${}^{A}R_C = {}^{A}R_C\, {}^{C}R_B$$

Interpretaciones de

1. Rota el punto ***P*** de {C} a {B} y luego de {B} a {A}.
2. Inicia con el sistema {A}, lo hace coincidente con {B} y luego con {C}

Álgebra lineal y sus aplicaciones en robótica con MATLAB

DÍA 2

- Del Piero Flores
- Diego Palma

Contenido

1. Transformaciones homogéneas (Traslación y rotación)
2. Composición de transformaciones homogéneas
3. Cinemática de un robot
4. Transformación proyectiva inversa
   a. Modelo de proyectividad de una cámara
5. Reto: Mapeo proyectivo inverso

Transformaciones homogéneas

- Representan la posición y orientación ("pose") de un sistema de referencia con respecto a otro sistema de referencia.

Matriz de Rotación (orientación)

Traslación (posición)

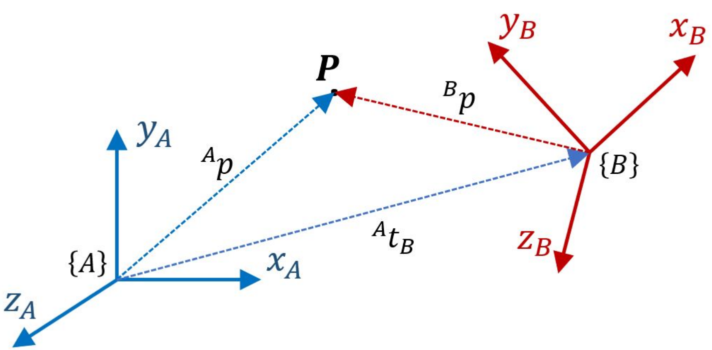
$$T = \left[\begin{array}{ccc|c} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ \hline 0 & 0 & 0 & 1 \end{array}\right] = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{bmatrix}$$

Ejemplo:

$$Trasl_x(d) = \begin{bmatrix} 1 & 0 & 0 & d \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Trot_z(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

¿Por qué utilizar las transformaciones homogéneas?



Representar ***P*** del sistema ***{B}*** al sistema ***{A}***: ${}^{A}p = {}^{A}t_B + {}^{A}R_B\,{}^{B}p$

Nos permite representar las ecuaciones en forma más compacta:

$$\underbrace{\begin{bmatrix} {}^{A}p \\ 1 \end{bmatrix}}_{{}^{A}\tilde{\boldsymbol{p}}} = \underbrace{\begin{bmatrix} {}^{A}R_B & {}^{A}t_B \\ \boldsymbol{0} & 1 \end{bmatrix}}_{{}^{A}\boldsymbol{T}_{\boldsymbol{B}}} \underbrace{\begin{bmatrix} {}^{B}p \\ 1 \end{bmatrix}}_{{}^{A}\tilde{\boldsymbol{p}}}$$

Coordenada homogénea

Matriz de transformación homogénea

Coordenada homogénea

Composición de transformaciones homogéneas

Es similar a la composición de rotaciones. Se sigue las dos siguientes reglas:

- Cuando se aplica una transformación con respecto al sistema fijo: Se **pre**-multiplica.
- Cuando se aplica una transformación con respecto al sistema actual: Se **post**-multiplica.

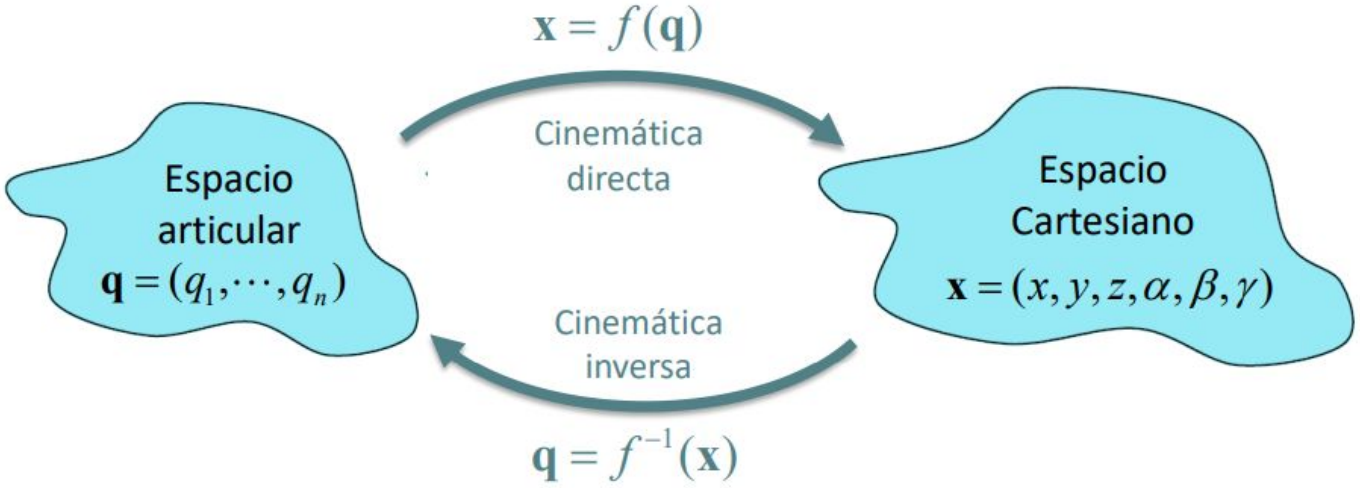
Ejemplo:

Encontrar la matriz de transformación homogénea que representa lo siguiente:

- Una rotación de un ángulo α alrededor del eje z,
- Seguido de una traslación de b unidades a lo largo del nuevo eje y,
- A continuación, se realiza una traslación de d unidades a lo largo del nuevo eje z,
- Finalmente, una rotación de un ángulo θ alrededor del nuevo eje z.

Veamos cómo solucionarlo en MATLAB

Cinemática de robots manipuladores

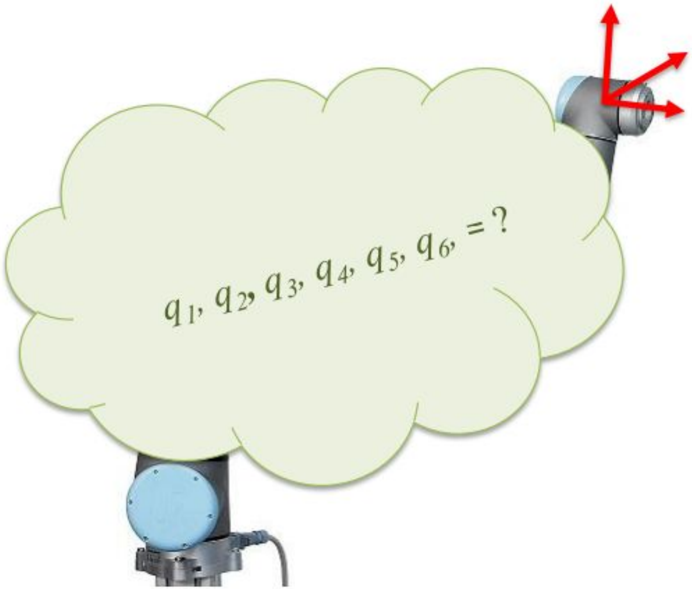


Cinemática directa: Dada una configuración articular, determina la posición/orientación de alguna parte del robot. (ejem. Efector final)

Cinemática inversa: Dada una configuración articular, determina la posición/orientación de alguna parte del robot. (ejem. Efector final)

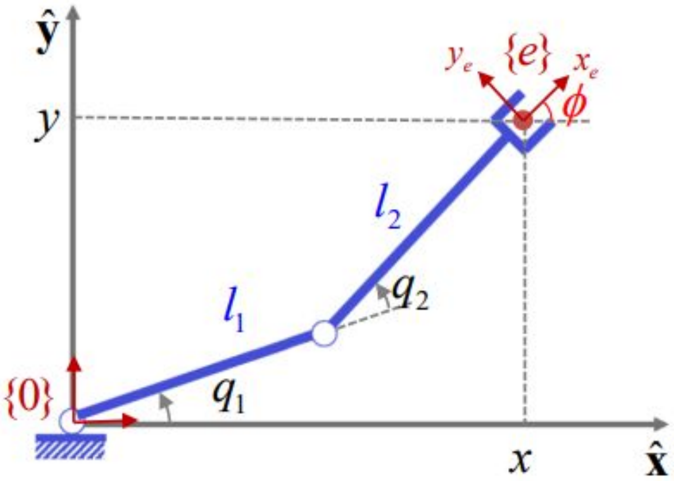
Formulaciones

Cinemática directa

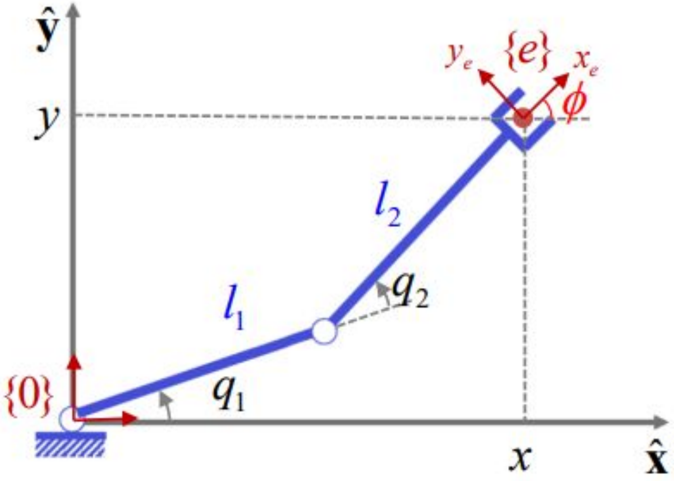


Cinemática inversa

Cinemática de un robot R-R



Cinemática de un robot R-R



- Posición del efector final

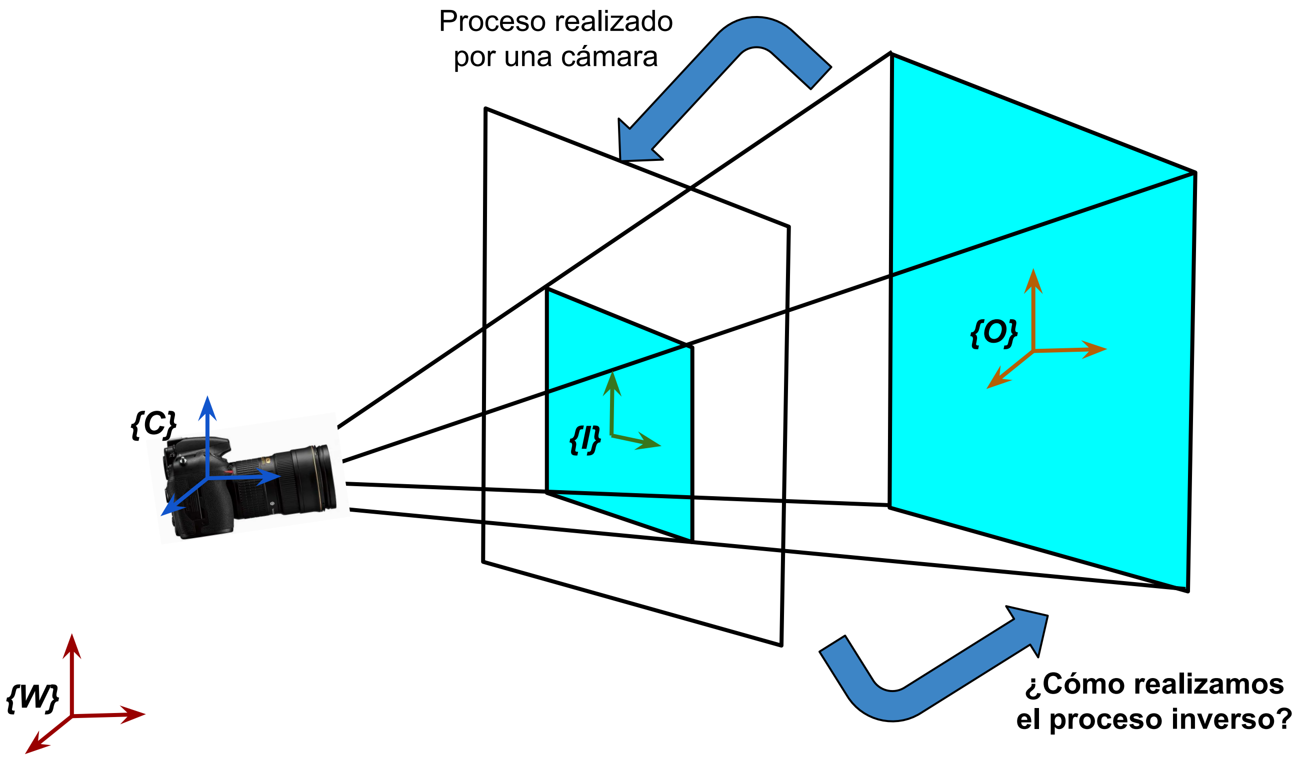
$$x = l_1 \cos(q_1) + l_2 \cos(q_1 + q_2)$$

$$y = l_1 \sin(q_1) + l_2 \sin(q_1 + q_2)$$

- Orientación del efector final

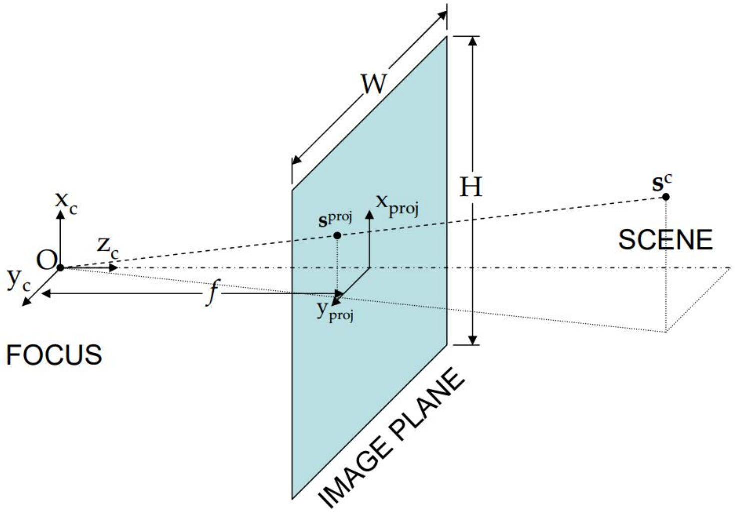
$$\phi = q_1 + q_2$$

Transformación Proyectiva Inversa



Es necesario tener una imagen RGB, la información de profundidad y **encontrar la transformación entre ambos sistemas de coordenadas (conocer los parámetros intrínsecos de la cámara)**.

Modelo de proyectividad de una cámara



Este modelamiento describe las matemáticas de la transformación de un punto del entorno a un punto de la imagen. Matemáticamente, esta relación se expresa como ***(u, v)=f(X,Y,Z)***, o de forma más específica como:

$$p = K[R|t]P$$

$$s\begin{bmatrix} u \\ v \\ Z \end{bmatrix} = \begin{bmatrix} f_x & \gamma & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Recordemos que nuestro objetivo es poder reconstruir el entorno, por lo que es necesario encontrar los puntos [X, Y, Z]. Consideramos:

- Un factor de escalamiento igual a 1.
- Visualizamos el entorno desde un sistema de coordenadas ubicado en la cámara. Esto significa que la matriz de rotación es igual a la matriz identidad y que el vector de traslación es un vector de ceros.

Por ende, la ecuación queda como sigue:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} Z = K \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

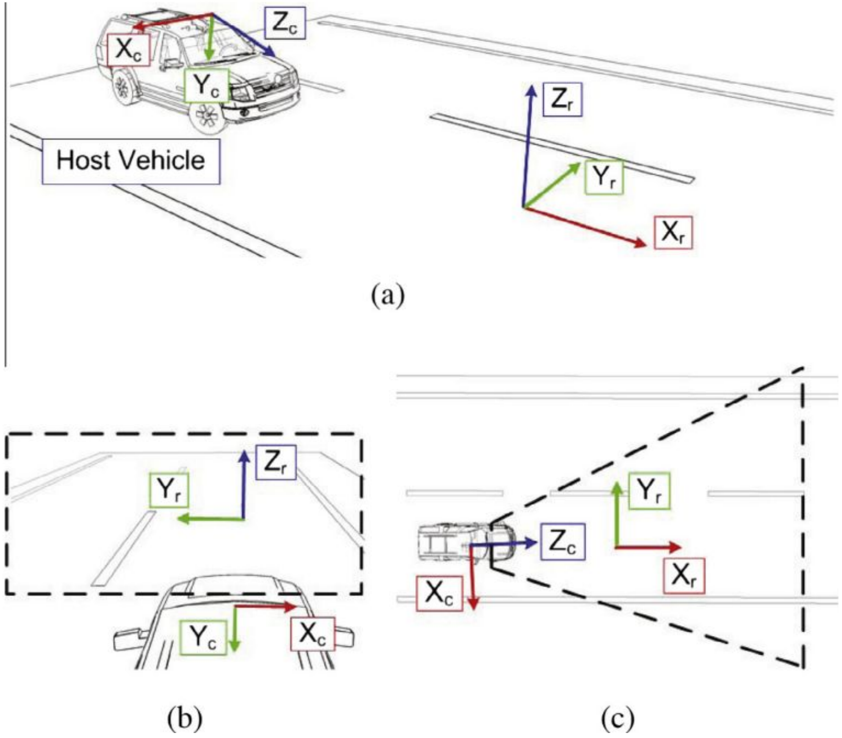
Finalmente, despejamos y se obtiene:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = K^{-1} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} Z$$

Veamos cómo codearlo en MATLAB

Mapeo proyectivo inverso

¿Qué pasa si no conocemos la información de la profundidad, pero quisiéramos ver la imagen desde otra perspectiva?



¿Puedes resolverlo?

¡Gracias!